Empfehlungssysteme

Andreas Kaspari

04.07.2006

Motivationen für die Entwicklung

- Überangebot an Information
- Immer grössere Anzahl an Internetteilnehmern
- Austausch von Wissen innerhalb von Gruppen mit ähnlichen Interessen und Vorlieben
- aber auch: Gewinnmaximierung durch gezielte Werbung und personalisierte Dienste
- Grundproblem ist eines der Grundlegendsten der Informatik: die Nadel im einem exponentiell wachsenden Heuhaufen zu finden

Empfehlungssysteme im Internet

- Bücher, CDs, Filme: amazon.com
- Filme: movielens
- WWW, Usenet: phoaks, grouplens
- **Musik**: last.fm, pandora, musicsurfer

Geschichte

- Ursprünge des Forschungsgebiets: cognitive science, approximation theory, information retrieval, forecasting theory, management science, consumer choice modeling
- Eigenes Forschungsgebiet seit Mitte der 90er Jahre.
- Dann Fokus auf Empfehlungen mit expliziten Gegenstandsbewertungen der Benutzer
- Erste Arbeiten verwendeten reines kollaboratives Filtern
- Inhaltsbasierte Methoden (inspiriert durch IR-Forschung), daher erste Systeme für Text (Artikel, Webseiten,...)

Zwei grosse, komplementäre Ansätze:

- Collaborative Methods
  - Basis sind z.B. Bewertungen, z.B. binär (gekauft/nicht gekauft) oder auf einer Skala von 1 bis 10
  - kein Bezug zum Inhalt der Gegenstände, daher vielseitig einsetzbar
- Content-based Methods
  - Basis sind Attribute der Gegenstände, z.B. Audiofeatures bei Musik, häufige Worte (TF/IDF) bei Text,...
  - daher starke Spezialisierung auf einen Medientyp

Zusätzlich:

- Hybride Methoden
  - verwenden Verfahren aus beiden Ansätzen
  - Kombination der Vorteile

Kollaboratives Filtern (CF)

- Benutzer bewerten einen Bestand von Gegenständen (z.B. Musikstücke, Fotos, Websites, Forschungsartikel usw.)
- Menschen, die z.B. die viele verschiedene Musikstücke ähnlich bewerten, haben auch bei unbekannten Musikstücken einen ähnlichen Geschmack
- Demjenigen, der viele meiner CDs mag, schenke ich Vertrauen, wenn er mir eine neue CD empfiehlt
- Bewährtes Standardverfahren. Wird kommerziell mit großen Erfolg beispielsweise von amazon eingesetzt.

Kollaboratives Filtern (CF)

Unterscheidung zwischen:

- Heuristische Verfahren (alle Verfahren, die kein Modell von der Problemdomäne bildet)
- Modellbasierte Verfahren (bilden beispielsweise ein statistisches Modell, welches Wahrscheinlichkeiten für bestimmte Bewertungen von Gegenständen angibt)

Heuristikbasiertes CF (aus [Adomavicius and Tuzhilin(2005)])

Sei C die Menge aller Benutzer, S die Menge aller Gegenstände und R die Menge aller möglichen Bewertungen. Dann ist die Nützlichkeitsfunktion u definiert als:

$$u : C \times S \rightarrow R$$

- u ist nur partiell definiert
- Extrapolieren auf den gesamten Bereich $C \times S$
- Definierter Bereich meist sehr klein im Verhältnis.

Heuristikbasiertes CF

- u ordnet jedem Gegenstand s im Kontext des Benutzers c eine Nützlichkeit zu.
- Einige Zuordnungen sind benutzerdefiniert
- Der Großteil wird vom Empfehlungssystem geraten.
- Empfehlung für den besten Gegenstand.

$$S_c^* = argmax_{s \in S} u(c, s)$$

- Mit S_c^* definiere Top-N-Liste für Benutzer c

Heuristikbasiertes CF

Wie wird u nun extrapoliert?

- Bewertungen anderer Benutzer sind bekannt
- Nützliche Bewertungen stammen von ähnlichen Benutzern
- Bestimme Menge $\hat{C} \subset C$ ähnlicher Benutzer mit $\hat{C} = N$
- Sei nun s von c unbewertet, dann ist die generierte Bewertung:

$$r_{c,s} = aggr_{c' \in \hat{C}} r_{c',s}$$

wobei $aggr : R^N \rightarrow R$

Heuristikbasiertes CF

Was ist *aggr*?

- Aggregiert alle einfließenden Bewertungen zu einem Wert
- Beliebig definierbar, aber einige gebräuchliche Definitionen.
- einfacher Durchschnitt:

$$r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s}$$

- Ist aber wenig aussagekräftig!

Heuristikbasiertes CF

- Allgemein gebräuchlich ist die gewichtete, normalisierte Summe:

$$r_{c,s} = k \sum_{c' \in \hat{C}} sim(c, c') r_{c',s}$$

wobei $sim : C \times C \rightarrow \mathbb{R}$ Ähnlichkeitsmaß für Benutzer und k Normalisierungsfaktor.

Heuristikbasiertes CF

- Problem: Benutzer benutzen Bewertungsskala unterschiedlich !
- Betrachte also die Abweichungen:

$$r_{c,s} = \overline{r_c} + k \sum_{c' \in \hat{C}} sim(c', c)(r_{c',s} - \overline{r_{c'}})$$

wobei $\overline{r_c}$ $\varnothing$-Rating von Benutzer c

Heuristikbasiertes CF

Distanzmaß $sim(c, c')$

- misst Ähnlichkeit von Benutzern
- dient der Bestimmung von $\hat{C}$
- gewichtet Bewertungen nach Ähnlichkeit der Benutzer

Definition von $sim(c, c')$ i.A. auf Grundlage der gemeinsam bewerteten Gegenstände:

$$S_{c_1 c_2} = \{s \in S \mid r_{c_1,s} \neq \emptyset \wedge r_{c_2,s} \neq \emptyset\}$$

Beliebteste Ansätze: Korrelations- oder Cosinus-basiert.

Heuristikbasiertes CF

Cosinus-basierter Ansatz:

- Bewertungen zweier Benutzer auf $S_{c_1 c_2}$ sind zwei m-dimensionale Vektoren im R^m.
- Bestimme den Winkel zwischen den Vektoren
- Kleinerer Winkel entspricht ähnlicheren Bewertungen und damit ähnlicheren Benutzern

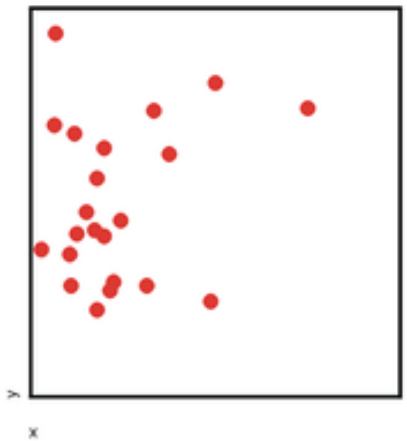
$$sim(c_1, c_2) = cos(\vec{c_1}, \vec{c_2}) = \frac{\vec{c_1} \cdot \vec{c_2}}{\|\vec{c_1}\|_2 \times \|\vec{c_2}\|_2}$$

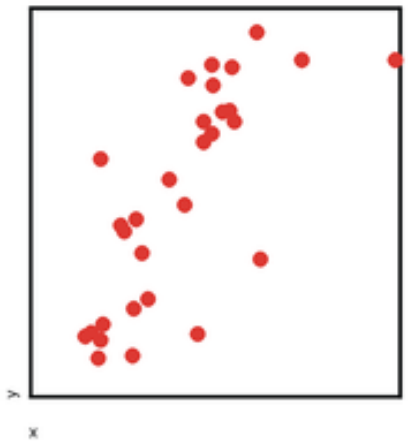
Wenn $\measuredangle(\vec{c_1}, \vec{c_2}) \rightarrow 0$, dann $cos(\vec{c_1}, \vec{c_2}) \rightarrow 1$

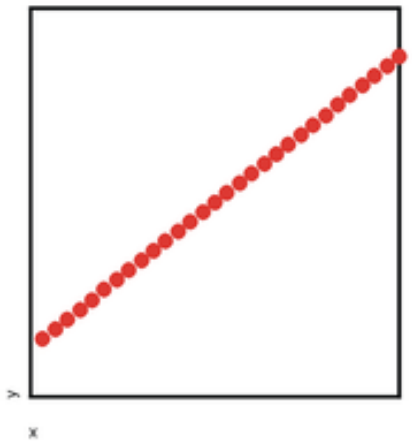
Heuristikbasiertes CF

Korrelationsbasierter Ansatz[Shardanand and Maes(1995)]:

$$sim(\vec{c_1}, \vec{c_2}) = \frac{\sum_{s \in S_{c_1 c_2}} (r_{x,s} - \overline{r_x})(r_{y,s} - \overline{r_y})}{\sqrt{\sum_{s \in S_{c_1 c_2}} (r_{x,s} - \overline{r_x})^2 \sum_{s \in S_{c_1 c_2}} (r_{y,s} - \overline{r_y})^2}}$$







Heuristikbasiertes CF

Item-to-item similarity[Karypis(2001)]:

- Definiere $sim(\cdot, \cdot)$ auf Gegenständen
- z.B. Cosinus zweier n-dimensionaler Vektoren im R^n (n entspricht Anzahl der Benutzer)
- alternativ: Korrelationsbasierte Ansätze,...
- ansonsten analoge Vorgehensweise

Heuristikbasiertes CF

Ähnlichste Benutzer finden:

- Speicherbasierte Verfahren
- meist KNN

Heuristikbasiertes CF

Probleme:

- Mindestanzahl (meistens ziemlich groß) von Benutzern muss erreicht sein.
- $S_{c_1 c_2}$ oft klein oder sogar leer. Daher Schwierigkeiten ähnliche Benutzer zu finden.
  - Lösungsansatz: Default-Ratings einführen

Heuristikbasiertes CF

- **Neue Gegenstände** werden nicht empfohlen, da fast niemand sie bisher bewertet hat.
- **Neue Benutzer** müssen erst eine Menge von Gegenständen bewerten, bis das ES sinnvolle Empfehlungen geben kann.
- Ungewöhnlicher Geschmack wird schlecht unterstützt, Mainstream überlagert alles andere.
- Sehr rechenaufwendig !!!
  - Lösungsansatz: periodische Vorberechnungen

Praxisbeispiel: last.fm

- Gegenstände sind Musikstücke
- Loggen jedes abgespielten Musikstücks auf einem zentralen Server (durch Winamp, Itunes, ... Plugins)
- Abspielhäufigkeiten und -zeitpunkte als implizite Bewertungen
- Empfehlungen auf Basis dieser impliziten Bewertungen mit standard heuristikbasiertem CF (wöchentlich aktualisiert)

Modellbasiertes CF

Kritik an heuristikbasiertem CF[Hofmann(2003)]:

- Möglicherweise suboptimale Empfehlungen (Empirisch gestützt)
- Eigentlich wird nichts gelernt (Statistisches Modell, u.ä.) und die Problemdomäne daher nicht verstanden
- Empfehlungsgabe ist nicht nachvollziehbar
- Skalierungsprobleme: Hohe Platz- und Zeitkomplexität (generelles Problem von memory-based methods)
- Kaum anpassbar auf spezielles Problem
- Benutzerprofile werden gespeichert (Datenschutz?)

- CF als Klassifikationsaufgabe
- Betrachte eigene Bewertungen, z.b. auf diskreter Skala (1-10) als Class-Labels
- Ratings der anderen Benutzer bilden Featurevektoren

	E_1	E_2	E_3
U_1like	*1*	*0*	*1*
U_1dislike	*0*	*0*	*0*
U_2like	*0*	*0*	*0*
U_2dislike	*0*	*1*	*0*
U_3like	*1*	*1*	*0*
U_3dislike	*0*	*0*	*1*
Class	*like*	*dislike*	*dislike*

- Lerne für jeden Benutzer ein Klassifikationsmodell auf Basis vorhanderer Bewertungen
- Empfehlungsgabe ist dann Einklassifizieren eines neuen Featurevektors in eine der Bewertungsklassen
- Viele Machine Learning Verfahren anwendbar (Neuronale Netze, SVMs, Decision Trees,Bayesche Netze)

Ansatz von [Billsus and Pazzani(1998)]

- Transformation in binäre Featurevektoren (auch mehrstufigen Bewertungsskalen)
- Verringerung der Dimension der Vektoren (SVD)
- Training eines Neuronalen Netzes auf reduzierten Vektoren
- Trainiertes Neuronales Netz assoziiert neuen Eingabevektor nach SVD mit entsprechender Bewertungsklasse

Inhaltsbasierte Methoden

- Gegenstände werden durch Features repräsentiert
- Beispielsweise Genre,Titel,... bei Filmen, Audiofeatures bei Musik oder die x häufigsten Worte bei Text
- Wiederum Nützlichkeitsfunktion:

$$u(c, s)$$

für unbekannten Gegenstand s.

- Wird auf Grundlage der Wertungen $u(c, s_i)$ berechnet, wobei $s_i \in S$ inhaltlich ähnlich zu s sind.
- Dann Empfehlungen für Gegenstände, die ein hohes Maß an inhaltlicher Ähnlichkeit zu den Preferenzen des Benutzers haben.

Sei S die Menge der Gegenstände, $s \in S$ und $Content(s) \in F^n$ die Menge der Featurewerte, die den Gegenstand s charakterisieren.

Sei C die Menge aller Benutzer, $c \in C$ und $ContentBasedProfile(c)$ die Geschmäcker und Preferenzen des Benutzers.

- Bei Text z.B. ein Gewichtevektor $(w_{c_1}, ..w_{c_k})$ für das Vorkommen bestimmter Schlüsselworte
- Oder Modell für einen binären Klassifizierer, z.B. bei Musik:

$$Class : F^n \rightarrow \{0, 1\}$$

Auswertung von $u(c, s)$

$$u(c, s) = score(ContentBasedProfile(c), Content(s))$$

Verschiedene Bedeutungen von *score*:

- cosine similarity measure $u(c, s) = cos(\vec{w}_c, \vec{w}_s)$ bei Text
- Korrelationsbasierte Ähnlichkeitsmaße
- Anwendung eines Klassifikators auf einen Featurevektor $x \in F^n$ bei Medien wie Musik.

Zeitaufwand

- Per Hand annotiert
- Featureextraktion bei Audio

Überspezialisierung

- Es kann nur Empfehlungen für Gegenstände, die ähnlich zu bereits bekannten sind, geben.
- Gegenstände, die zu ähnlich sind, will der Benutzer genauso wenig, wie Gegenstände die zu unähnlich sind.
- Möglichkeit: Zufallskomponente einfügen und sehr ähnliche Gegenstände aus Empfehlungen herausfiltern

Praxisbeispiel: pandora

- Vorgegebene Musikbibliothek
- Alle Musikstücke sind per Hand annotiert worden
- Binäre Features, wie z.B. orchestral arranging, chromatic harmony, electric instrumentation, basic rock song structure (Verse, Chorus, Verse, Chorus,...)
- Binäre Bewertungen (Mag ich bzw. Mag ich nicht)

Arbeitsweise

- Erstellen eine sogenannten Station durch Angabe von Musikstück (oder ganzem Künstler)
- Dies initialisiert *ContentBasedProfile*
- Webradio spielt nun hintereinander zu *ContentBasedProfile* inhaltlich ähnliche Stücke
- Begründung mit Verweis auf die Features wird angezeigt
- Weitere Verfeinerung des Profiles durch $(0, 1)$-Bewertungen der vorgeschlagenen Musikstücke
- Geschlossenes System, Details wurden nicht veröffentlicht.

Hybride Methoden

Separate Systeme kombinieren

- Bewertungen werden linear kombiniert
- Abstimmungsverfahren
- Auswahl der übernommenen Bewertung durch *Qualitätsmetrik*

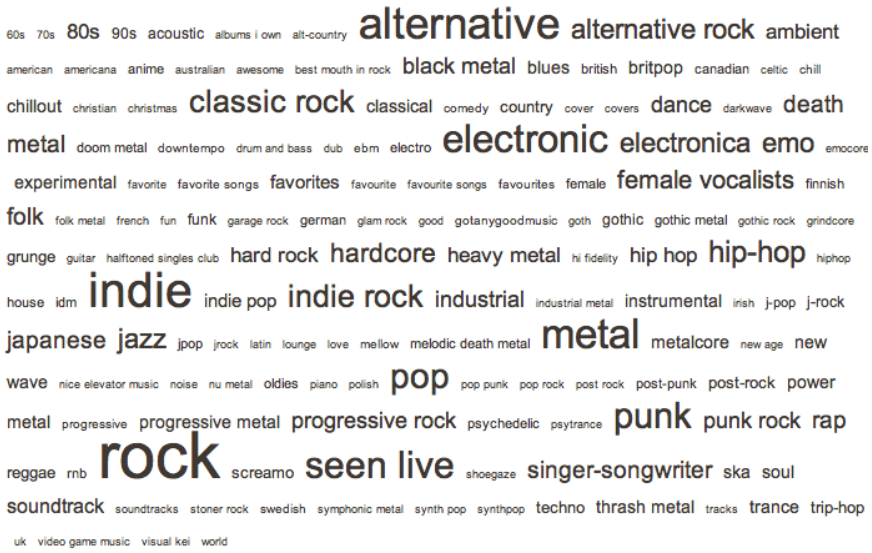
Inhaltsbasierte Meth. als Erweiterung für CF

- Inhaltsbasierte Profile für Benutzer
- Bestimmen von Nachbarn durch diese Profile (nicht ähnliche Bewertungen)
- behebt das Problem der *sparsity*
- zusätzlich auch direkte inhaltsbasierte Empfehlungen

Nemoz

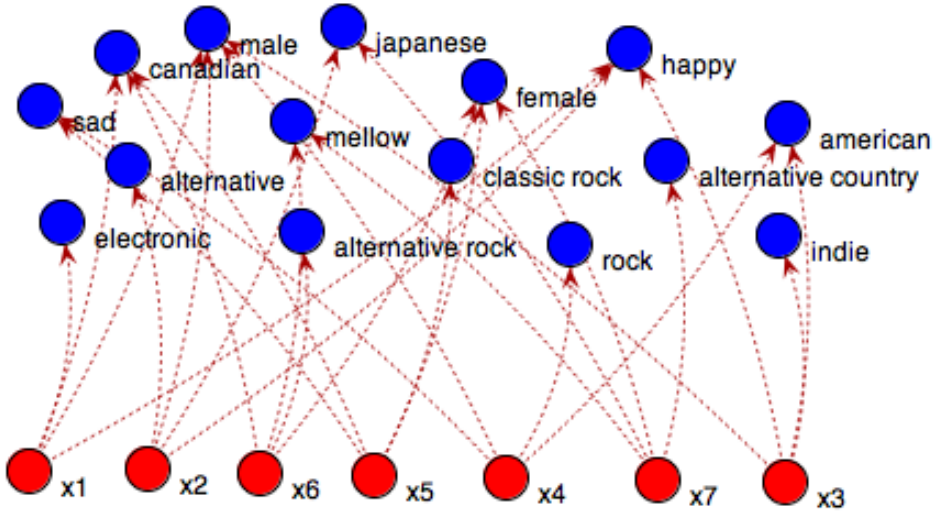
Web 2.0 tagging

- Benutzer weisen ihren Musikstücken Tags zu (bekannt aus den Web 2.0 Diensten flickr, del.icio.us oder last.fm)



- Problem: Unübersichtlichkeit nimmt schnell zu!

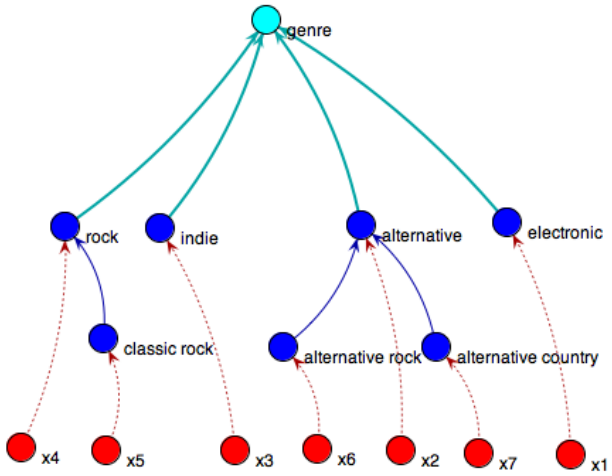
Web 2.0 tagging (2)

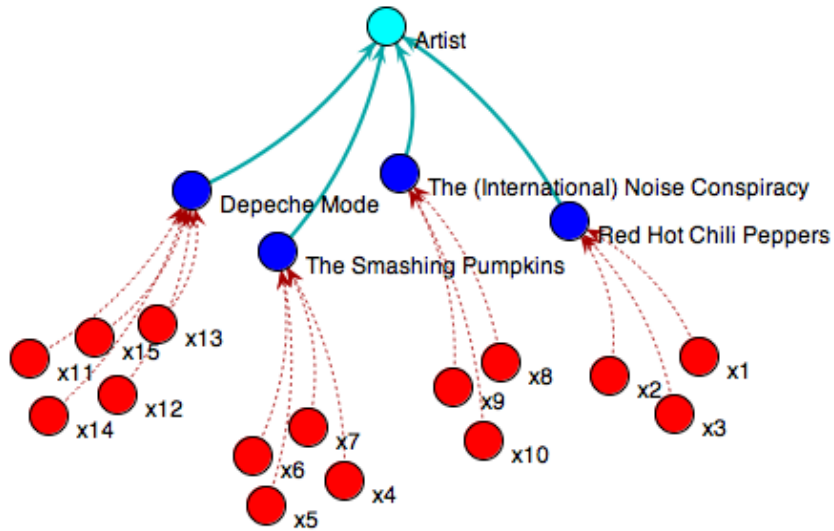


- keine sinnvolle Semantik
- nicht verwendbar für machine learning Methoden

Multi-aspect tagging

- Taghierarchien, um z.B. Subgenre-Genre-Beziehungen auszudrücken
- Aspekte für eine Gruppe von Tags (z.B. Genre und Artist)





Multi-aspect tagging (2)

- Features (Kollaborativ als auch inhaltsbasiert) für jedes Tag
- Entsprechende Feature-Werte für jedes Musikstück
- Erlaubt den Einsatz einer großen Bandbreite von Machine Learning Verfahren

Automatisches Tagging

- Benutzer annotiert kleinen Anteil seiner Musik mit Tags. Dies trainiert einen Klassifizierer.
- Diese Annotierung geschieht nach einem Aspekt (z.B. Stimmung). Tags unter diesem Aspekt bilden einen Baum.
- Automatische Annotierung per hierarchischer Klassifikation
- Anwendung: Browsen fremder Musik durch die eigene *Brille* bzw. Aspekt.
- Sozusagen: Halbautomatische Empfehlungen.

Kollaboratives Filtern

- Lässt sich in diesem Framework einfach ausdrücken
- Spezialfall des automatischen Taggings
- $sim(\cdot, \cdot)$ auf Benutzern beispielsweise durch ähnliche Strukturen
- Zusätzliche Einbeziehung von inhaltsbasierten Features (Hybridsystem)

Implementierung

- Wird realisiert in Nemoz (nemoz.sf.net)
- Musikmanager mit *intelligenten* Erweiterungen
- Ehemaliges PG Projekt an der Uni Dortmund
- Open-Source, geschrieben in Java
- Momentan eher ein Forschungsvehikel
- Weiterentwicklung u.a. in anstehenden Diplomarbeiten


Member-Gediminas Adomavicius and Member-Alexander Tuzhilin.
Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.
IEEE Transactions on Knowledge and Data Engineering, 17(6):734–749, 2005.
ISSN 1041-4347.
doi: http://dx.doi.org/10.1109/TKDE.2005.99.


Daniel Billsus and Michael J. Pazzani.
Learning collaborative information filters.
In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.
URL
citeseer.ist.psu.edu/billsus98learning.html.


Thomas Hofmann.
Collaborative filtering via gaussian probabilistic latent semantic analysis.
In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, New York, NY, USA, 2003. ACM Press.
ISBN 1-58113-646-3.
doi: http://doi.acm.org/10.1145/860435.860483.


George Karypis.
Evaluation of item-based top-n recommendation algorithms.
In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254, New York, NY, USA, 2001. ACM Press.

ISBN 1-58113-436-3.
doi: http://doi.acm.org/10.1145/502585.502627.

Upendra Shardanand and Patti Maes.
Social information filtering: Algorithms for automating “word of mouth”.
In *Proceedings of ACM CHI’95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.
URL `citeseer.ist.psu.edu/article/upendra95social.html`.